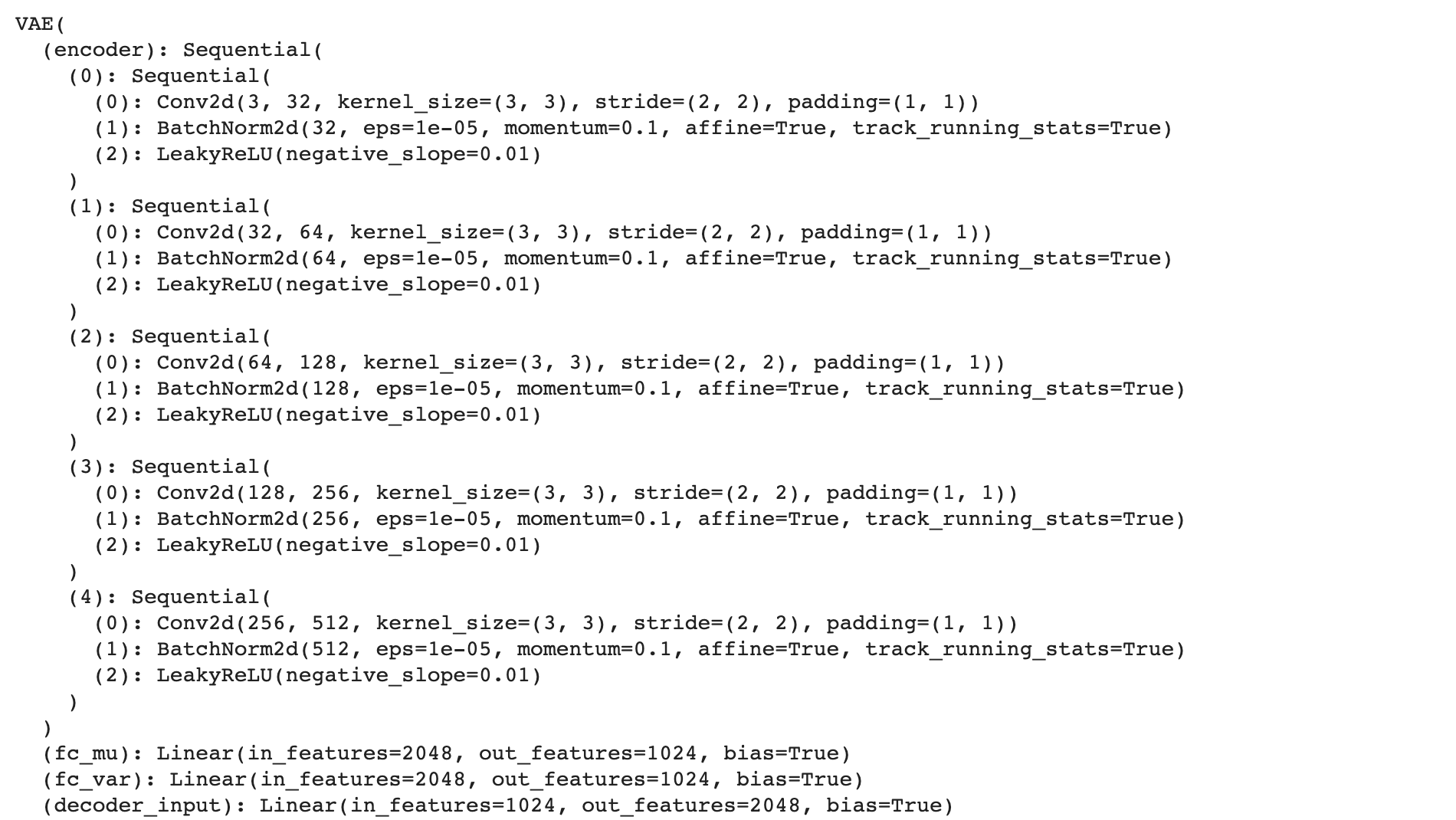
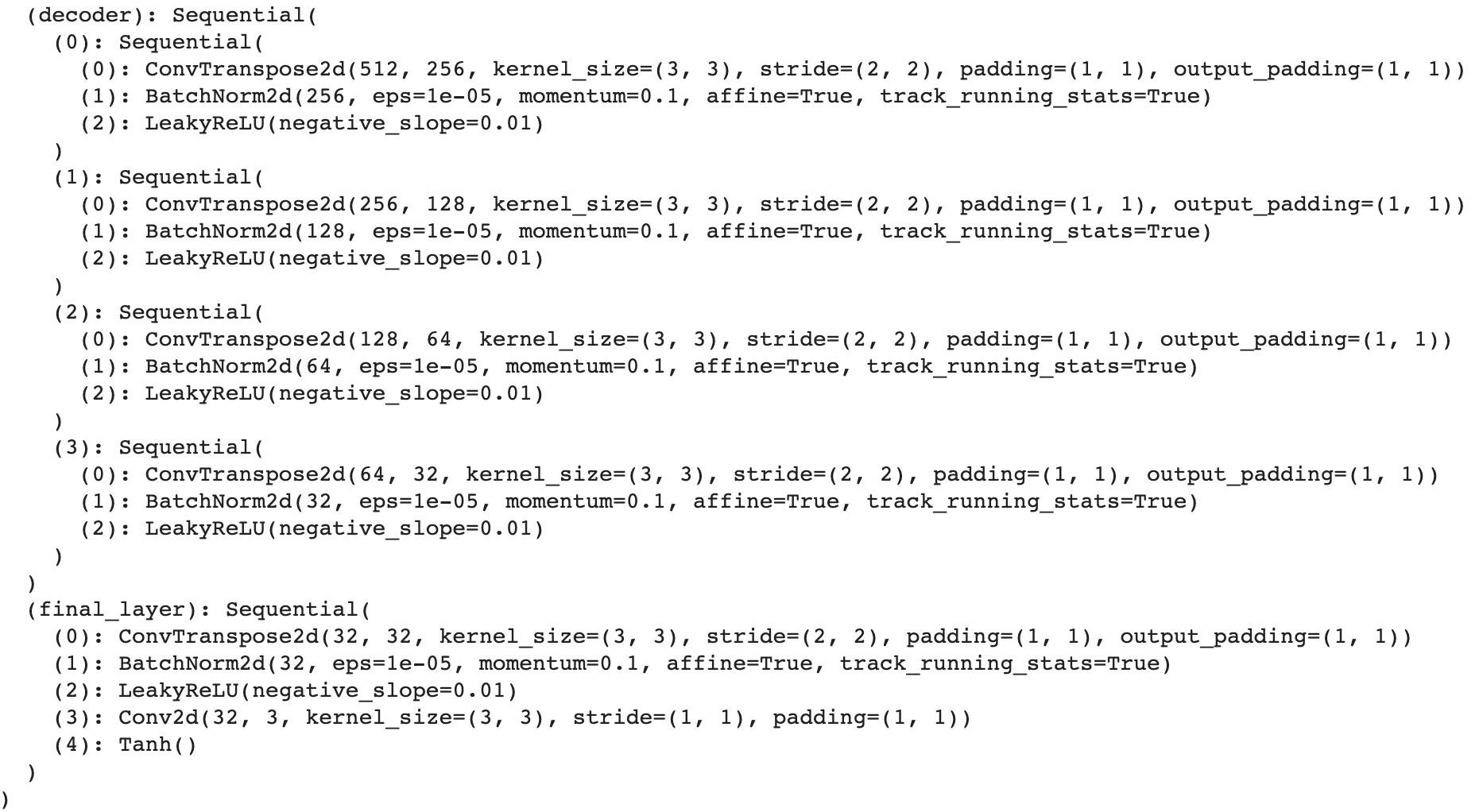
Hw3 report

姓名：傅敬倫 系級：電機四 學號：b06505011

Problem1. (30%) no collaboration

1.





我的VAE中分了encoder和decoder。Encoder中有五層conv，在latent space中的hidden size = 512，而decoder中有五層convtranspose。

Train：

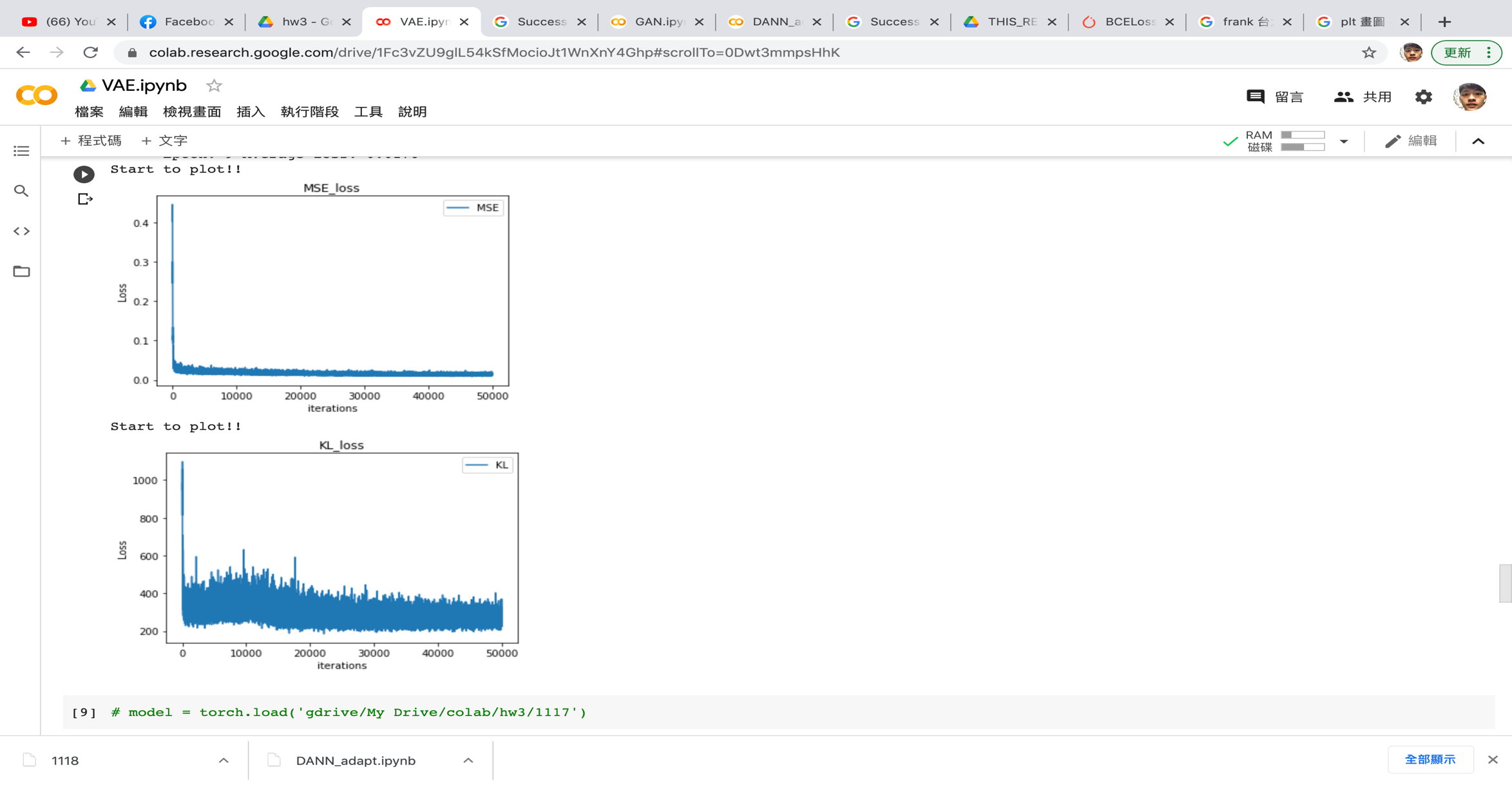
Optimizer : Adam

Learning rate : 0.0001, 每10個epoch下降一半

Loss : MSE and KL with weight 0.0001

2.

前50k steps的MSE\_loss和KL\_loss



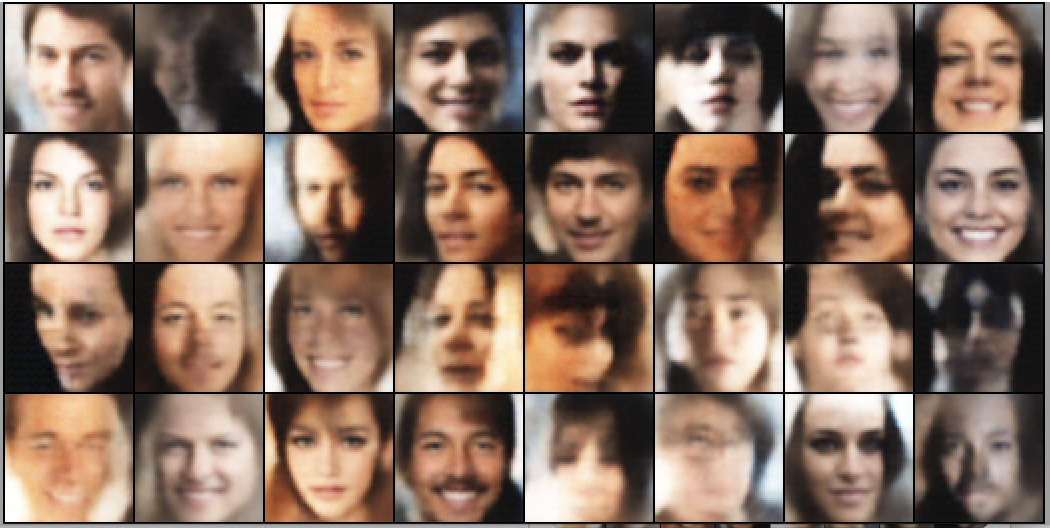
3.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test  Image |  |  |  |  |  |
| Recon.  Image |  |  |  |  |  |
| MSE | 0.0037 | 0.0037 | 0.0038 | 0.0033 | 0.0037 |

3.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test  Image |  |  |  |  |  |
| Recon.  Image |  |  |  |  |  |
| MSE | 0.0036 | 0.0027 | 0.0033 | 0.0033 | 0.0034 |

4.

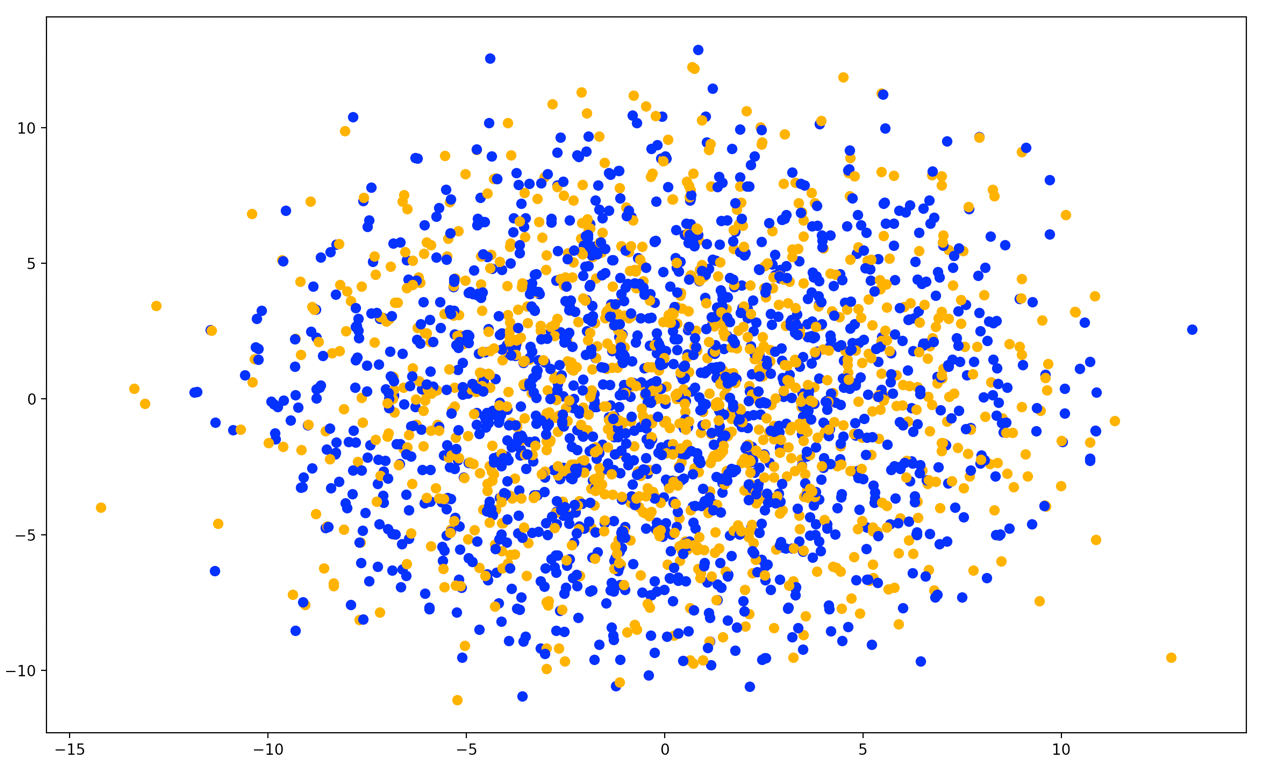


5.

藍色：女生 黃色：男生



藍色：not smiling 黃色：smiling



6.

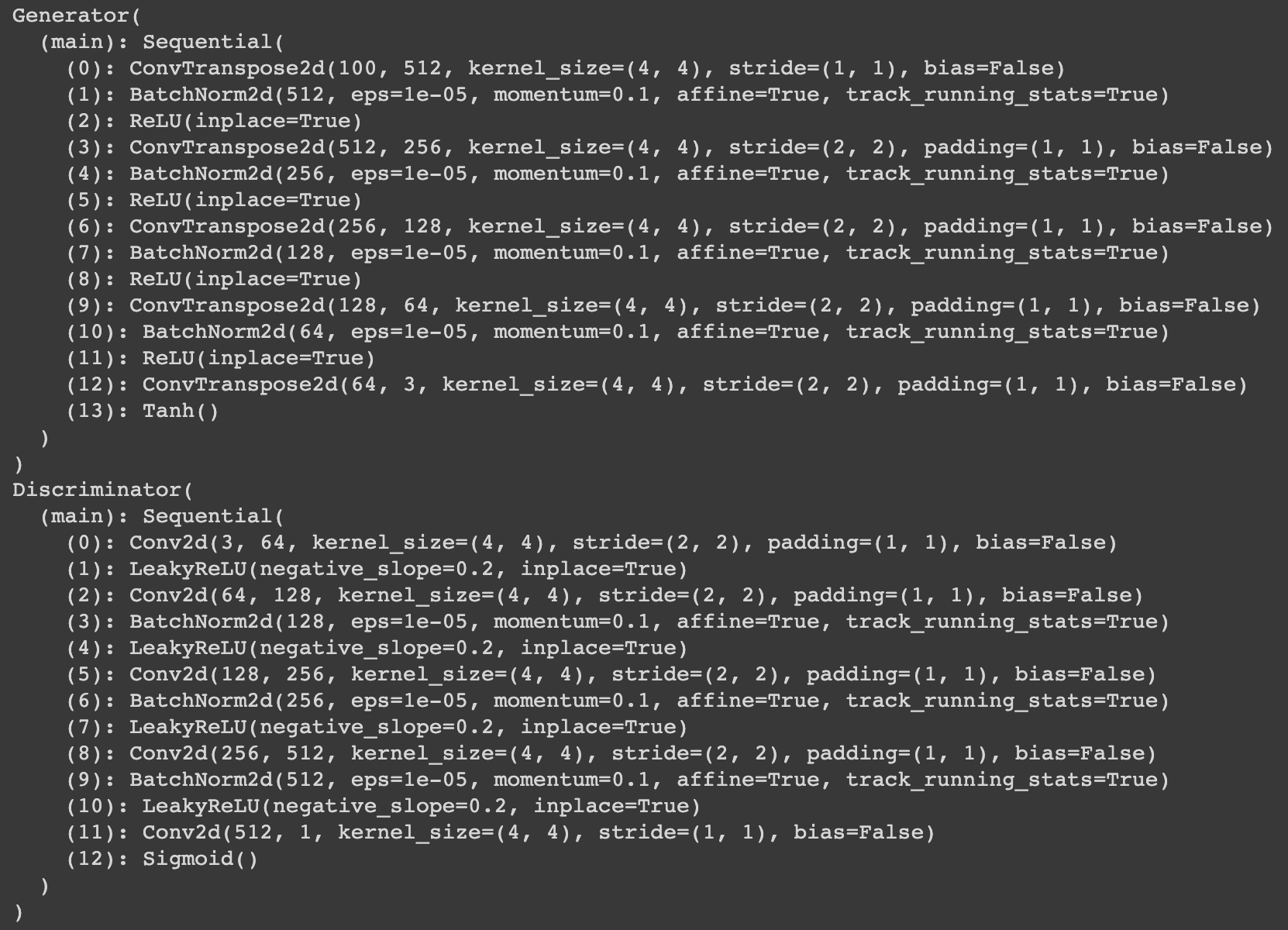
在train VAE時我有在每一個epoch把Sample出的圖片印出來（如下圖）：



這是前五個epoch我將目前model的sample圖片印出來，可以發現確實隨著reconstruct image的MSE越小，Sample的葡片也越好。但是在後面的epoch，我發現 reconstruct image的MSE越小，產生出的Sample不一定越好，甚至到後面還會越來越扭曲與模糊，因此我在train VAE時是每5個epoch會先看當前產生出的reconstruct image和sample。

Problem2. (20%) no collaboration

1.



Train :

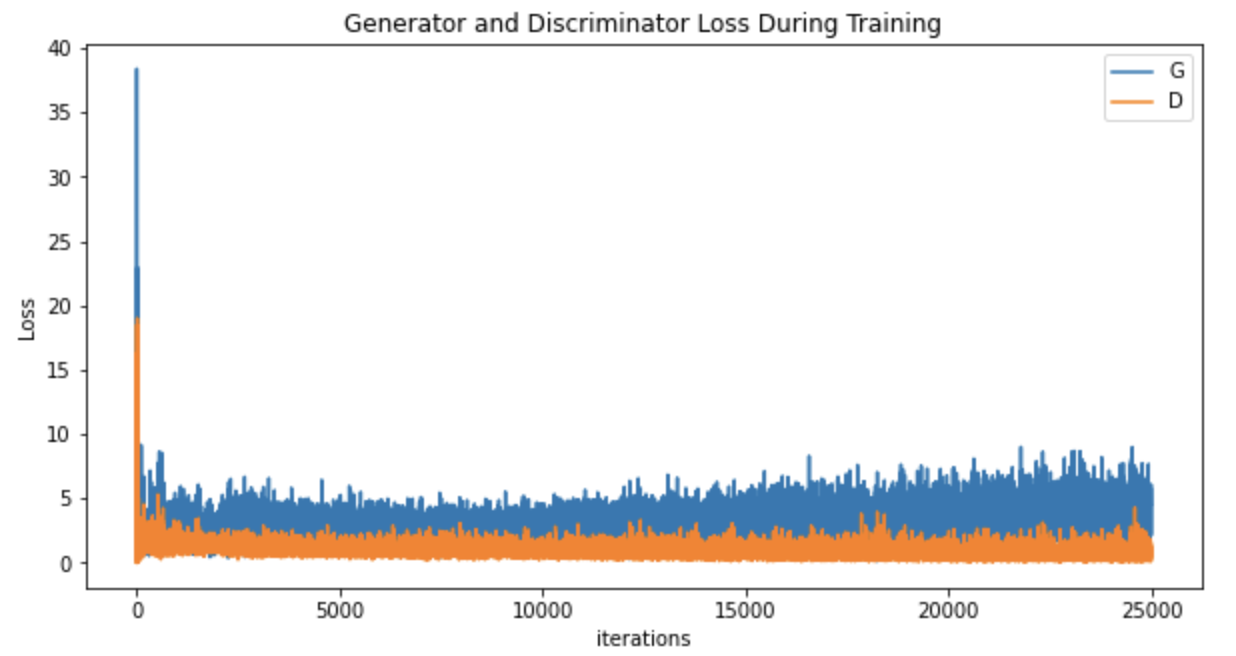
Optimizer : both Adam (G and D)

Learning rate : 0.001

Loss : BCE

Epoch : 10

這是在train時，G和D的loss



2.



3.

在訓練GAN的時候，Discriminator的初始值非常重要，很常如果都猜對的話就會完全train不起來，learning rate也不能調太大很容易就壞掉了。Train GAN不同於之前的model可能epoch不能太大，很容易在兩個model互相抗衡中爆掉，因此我GAN中的epoch只用了5，我也有試過epoch=10，其實兩者產生出的圖片蠻相似的，沒有特別好。

4.

在VAE產生出的圖片，可看到比較模糊，但還是看得出臉部的特徵，顏色也比較淡一點。在GAN產生的圖片中，整體的顏色更接近實際的膚色，更加亮麗，但是在臉部的特徵上大部分都有一些扭曲。

Problem3. (35%)

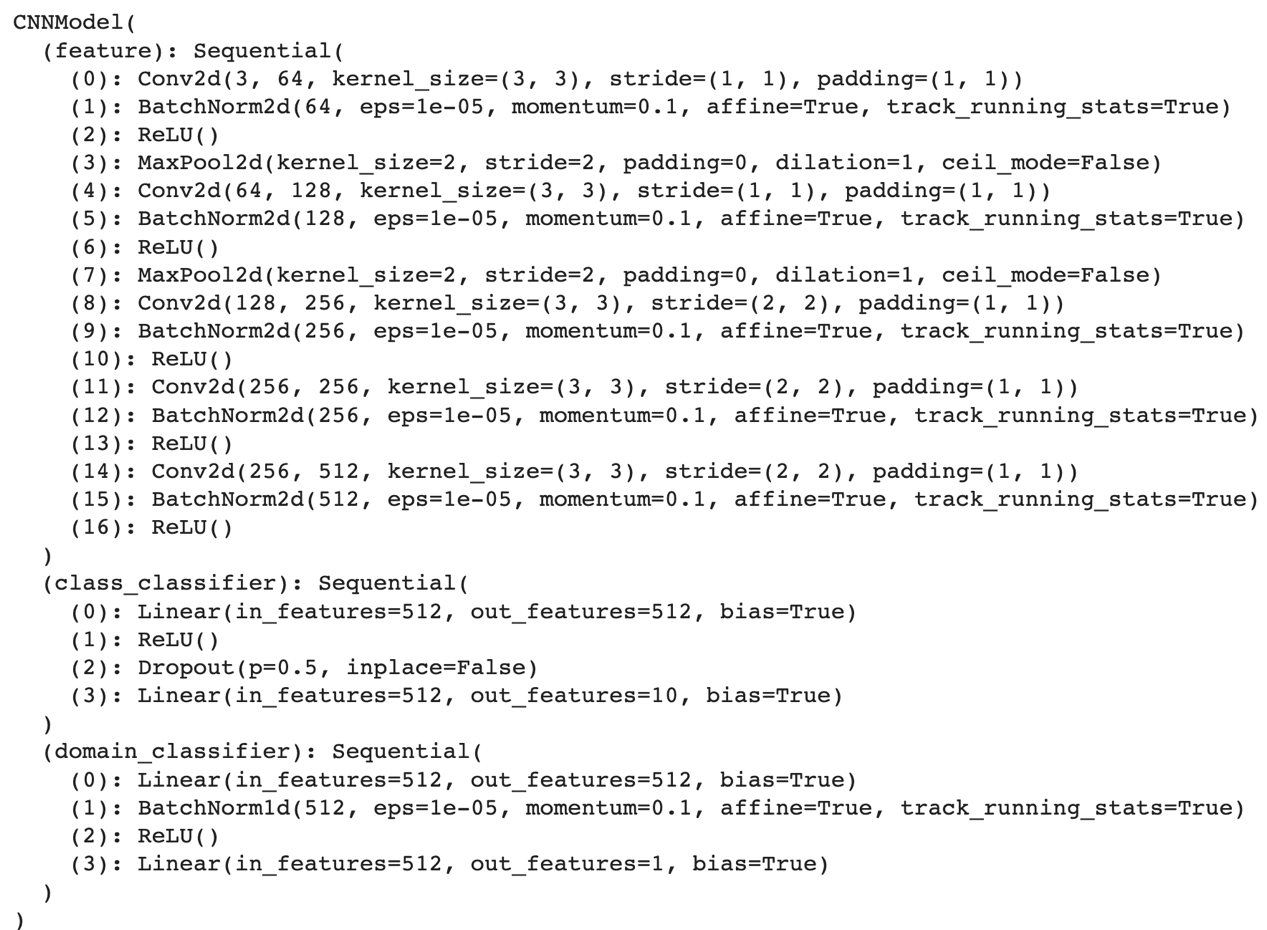
1.2.3.

|  |  |  |  |
| --- | --- | --- | --- |
|  | USPS -> MNIST-M | MNIST-M -> SVHN | SVHN -> USPS |
| Source only(im+label) | 20.8%  20 epoch | 34.4%  20 epoch | 68.9%  20 epoch |
| Source(im+label)  Target(im) | 61%  50 epoch | 41.9%  50 epoch | 72.1%  100 epoch |
| Target only(im+label) | 98.1%  20 epoch | 92.1%  20 epoch | 97.3%  20 epoch |

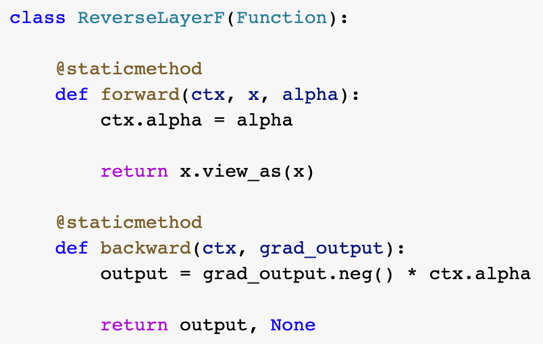
4.

|  |  |  |
| --- | --- | --- |
|  | label | domain |
| U 🡪 M |  |  |
| M 🡪 S |  |  |
| S 🡪 U |  |  |

5.



我的encoder中會有五個conv層，最後會將圖片降到512維 (feature size)，class\_classifier會將feature分到10個類別，domain\_classifier會將feature分到一個類別，因為我domain的loss是使用BCEWithLogitsLoss，在forward中我有用到ReverselayerF（如下圖）：



目的是讓model在backward時的gradient是用減的。

|  |  |
| --- | --- |
| Optimizer | Adam |
| Learning rate | 0.001 |
| Class\_loss | CrossEntropyLoss |
| Domain\_loss | BCEWithLogitsLoss |

6.

在訓練DANN時，我的Domain\_loss的參數不是像paper中寫的alpha會隨著epoch改變，而是都使用0.01。在三個任務中，第二個task（MNISTM🡪SVHN）是最難的，原本我的DANN的feature是512，但我發現在第二個中如果將feature加大成512\*2\*2正確率會提高非常多。

Problem4.

1.

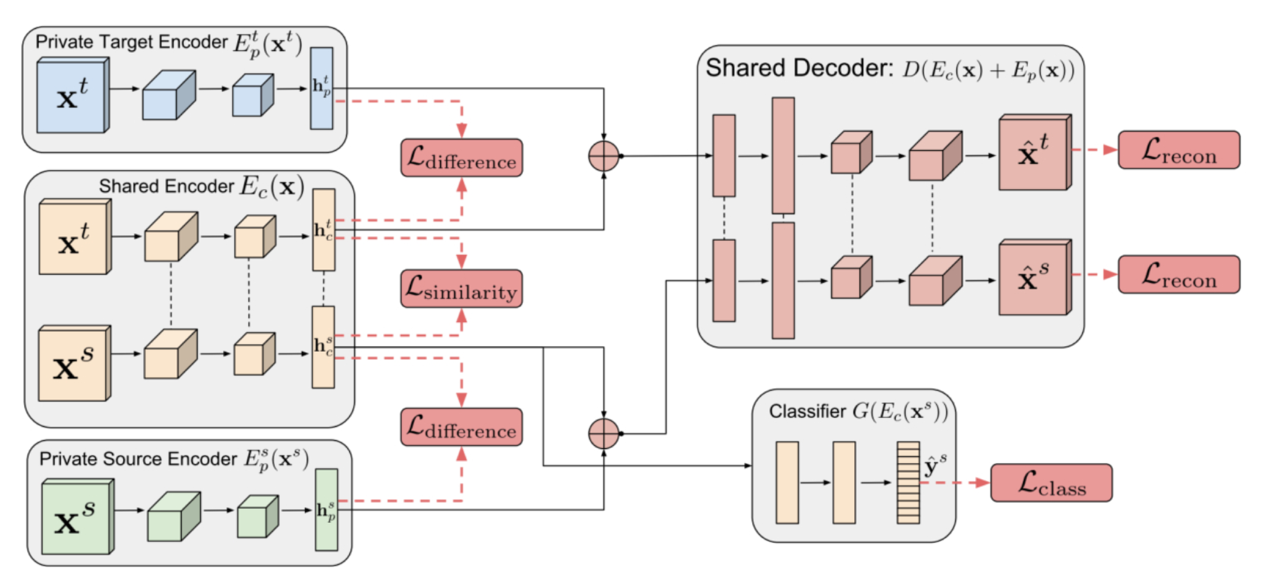
我使用的是DSN架構

|  |  |  |  |
| --- | --- | --- | --- |
|  | USPS -> MNIST-M | MNIST-M -> SVHN | SVHN -> USPS |
| DSN | 68%  100 epoch | 52.3%  50 epoch | 76.3%  100 epoch |
| DANN | 61%  50 epoch | 41.9%  50 epoch | 72.1%  100 epoch |

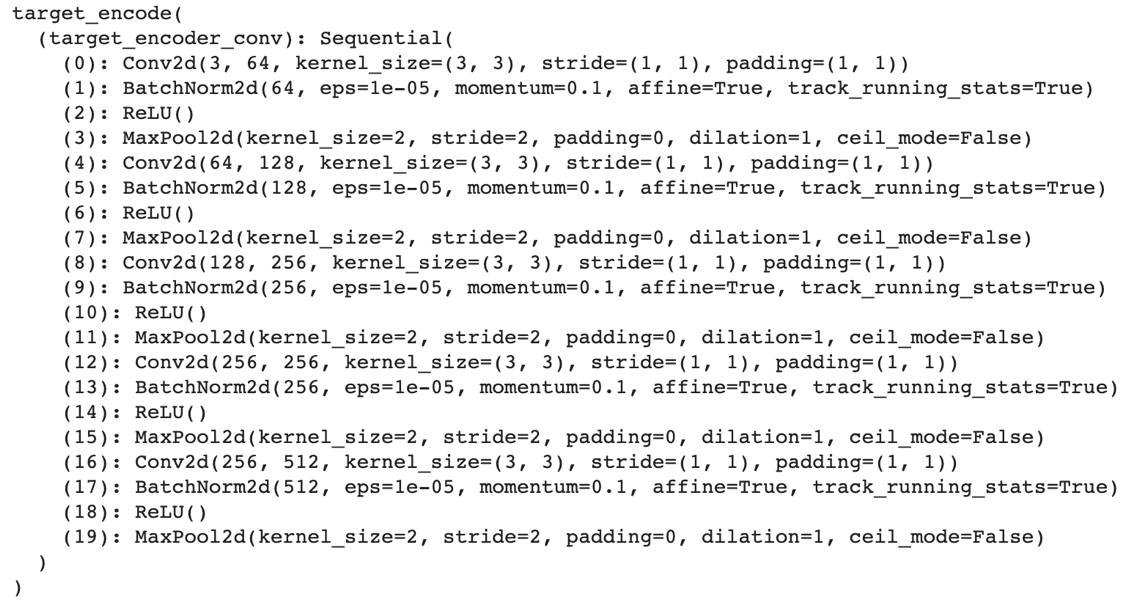
2.

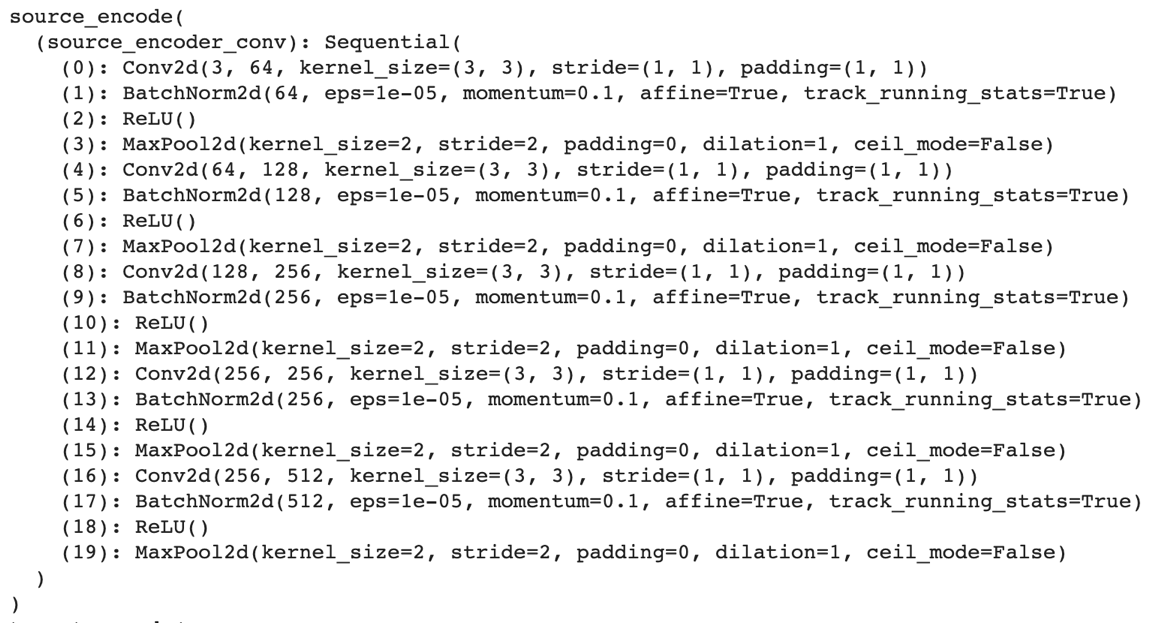
|  |  |  |
| --- | --- | --- |
|  | label | domain |
| U 🡪 M |  |  |
| M 🡪 S |  |  |
| S 🡪 U |  |  |

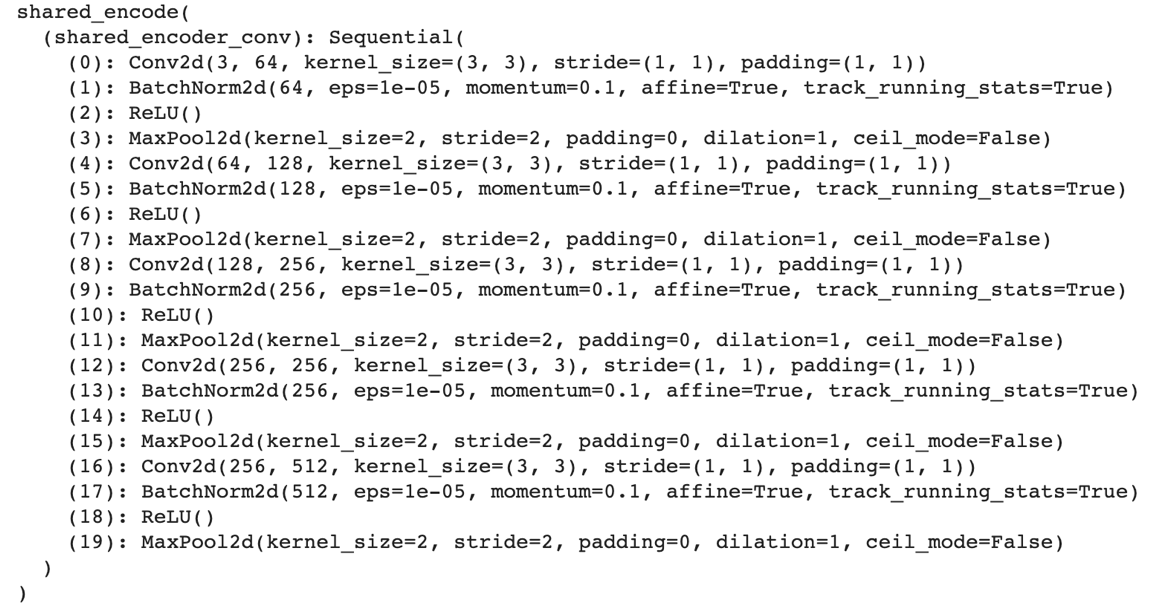
3.

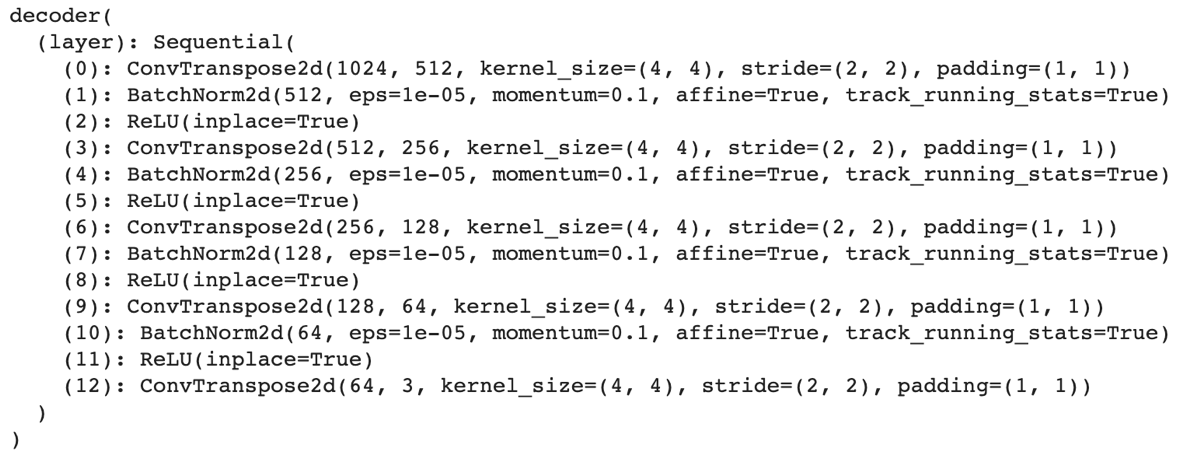


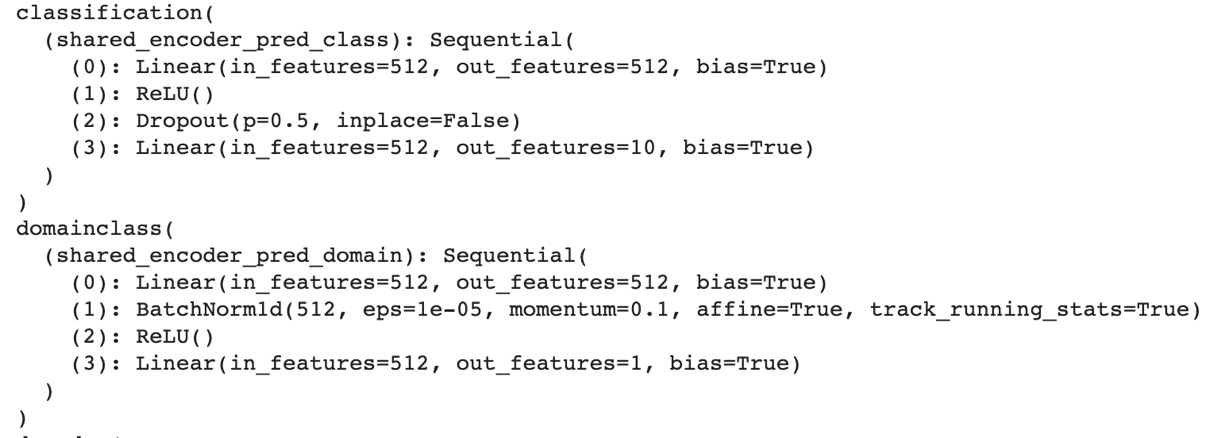
我的improve model是使用DSN架構（如上圖）。我將全部大致上分為六個model，分別為：Private Target Encoder, Private Source Encoder, Shared Encoder, Shared Decoder, Classifier, Domain Classifier





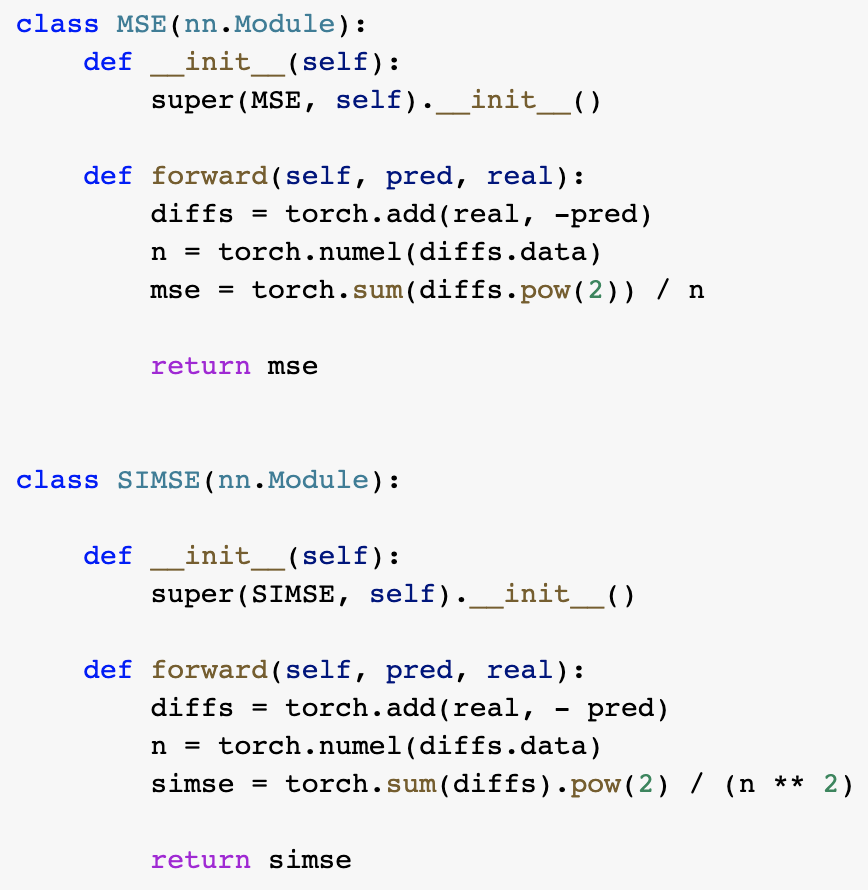






Train:

|  |  |
| --- | --- |
| Optimizer | Adam(6個model都是) |
| Learning rate | 0.0001 |
| Class\_loss | CrossEntropyLoss |
| Domain\_loss | BCEWithLogitsLoss |
| Recon\_loss | MSE( ) 自己定義的 |
| Sim\_loss | SIMSE( ) 自己定義的 |
| Diff\_loss | DiffLoss( ) 自己定義的 |



在train的時候，我是先train D(Domain Classifier)，讓D可以正確的分開Source和Target，接著我會訓練六個model就如同paper上的架構，這裡的Domain\_loss我是用減的，讓Domain分不清Source和Target，我每一項Loss的參數如下圖：



4.

在Train DSN時，其實比DANN來的難非常多，主要是太多Loss，而每個Loss的比例又很難抓，因此一開始train的時候，我先將Loss只用Class\_loss和Domain\_loss，Domain\_loss參數跟DANN的一模一樣都是0.1，觀察正確率是否會和DANN差不多。接著我再將Recon,Sim,Diff加入，去調三個的參數，為了方便三個都用依樣的倍率，最後我是使用0.01。在訓練時我發現Diff\_loss幾乎都等於0，learning rate在大於0.01是完全train不起來。在加入額外三個loss後，可以發現確實會比DANN來得更好一些，Private和shared之間的loss確實是有讓shared encode完的feature保留更多跨domain的資訊